

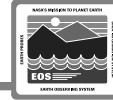
---

### Discussion Topics

One of the critical activities in testing the science code is to ensure that it conforms to certain standards. In this case we will test conformance to the ANSI standards.

**Briefly explain the ANSI standards: their origin, and why they are important.**

## Perform Standards Checks



- Checking ANSI Compliance FORTRAN 77
- Checking ANSI Compliance FORTRAN 90
- Checking ANSI Compliance C
- Checking ANSI Compliance Ada
- Checking Process Control Files
- Checking Metadata Control Files
- Checking for Prohibited Functions

520-TD-001-002

SSI&amp;T 5-2

## Discussion Topics

SCFs send code written in one of several languages. We want to make sure that before the code is integrated into the production system, it conforms to the ANSI standard.

In this lesson we will discuss in detail how to perform standards checking of the science software for each of the approved languages:


- FORTRAN 77
- FORTRAN 90
- C
- Ada

In addition, we want to check other files that are typically sent by the SCF to the DAAC in support of the science software.

The two typical files are:

- Process Control File (PCF)
- Metadata Control File (MCF)

## Checking ANSI Compliance FORTRAN 77



**Purpose:**

- To check that FORTRAN 77 language source files are compliant with the ANSI and allowed extensions.

**Tool:**

- FORCHECK

**Assumptions:**

- The FORTRAN 77 source files are available, accessible, and have read permissions.
- The environment variable FCKCNF has been set to the location of the FORCHECK configuration file.

520-TD-001-002

SSI&T 5-3

### Discussion Topics

**Purpose:** To check that FORTRAN 77 language source files are compliant with the ANSI and allowed extensions. The COTS used for this task is FORCHECK.

**Assumptions:** The FORTRAN 77 source files are available, accessible, and have read permissions. The environment variable FCKCNF has been set to the location of the FORCHECK configuration file.

#### References:

FORCHECK for Sun/SunOS, A Fortran Verifier and Programming Aid, Installation Guide

FORCHECK for Sun/SunOS, A Fortran Verifier and Programming Aid, User Guide

Help screen available through SSIT Manager


ANSI FORTRAN 77 Programming Language Standard, X3.9-1978

IEEE Standard 1003.9, FORTRAN77, Language Interfaces, Part 1: Binding for System Application Program Interface (API)

Data Production Software and Science Computing Facility (SCF) Standards and Guidelines, January 14, 1995 (CH-01 2/15/95), also on World Wide Web:

<http://spsosun.gsfc.nasa.gov/SCFStand.html>

### Checking ANSI Compliance FORTRAN 77 (cont'd)



- Select Tools from SSIT Manager
- Select FORCHECK from the Tools menu
- Specify global options
- Specify local options
- Examine output

520-TD-001-002

SSI&T 5-4

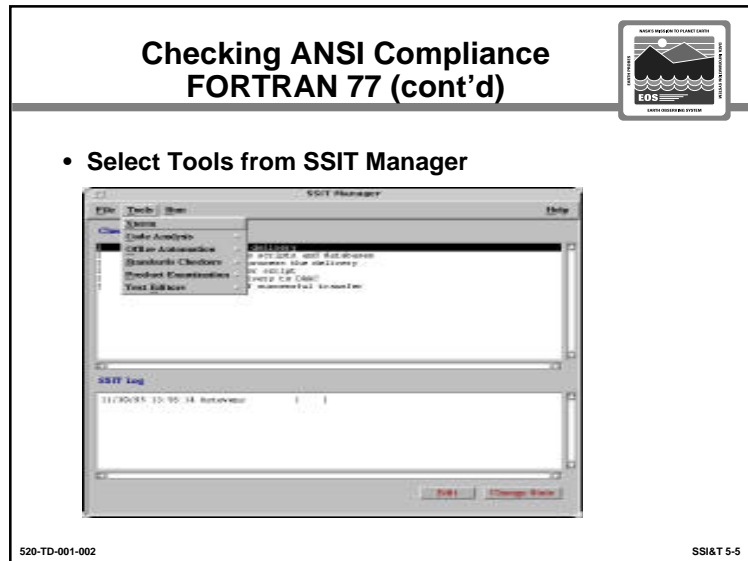
## Discussion Topics

**Note:** The FORCHECK program is run from the command line. However, it can be called from within the SSIT Manager (in interactive mode) which will create an xterm window within which to run the program. There is no GUI associated with FORCHECK.

FORCHECK can also be run in non-interactive mode from the command line. Refer to FORCHECK User Guide for a complete description.

The general procedure for checking ANSI compliance using the SSIT manager is as follows:

- Select Tools from SSIT Manager
- Select FORCHECK from the Tools menu
- Specify global options (optional)
- Specify local options (optional) and filename(s) to check
- Examine output



### Discussion Topics

#### Step 1. Invoke FORCHECK

- From the Tools menu of the SSIT Manager choose Standards Checkers. Then choose FORCHECK.


or

- From any available xterm on the Sun, enter `forchk`

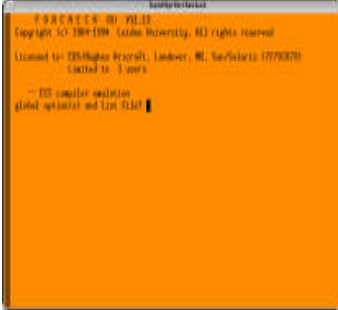
Choosing the first method will result in an xterm window (from the Sun) being displayed and the invocation of FORCHECK. The results in that xterm are the same as would occur by using the second method.

**NOTE:** FORCHECK is configured to check for the non-ANSI FORTRAN 77 extensions which are approved for use in ECS science code, according to "Data Production and SCF Standards and Guidelines." This implemented through the configuration file \$FCKCNF.

## Checking ANSI Compliance FORTRAN 77 (cont'd)



- Select FORCHECK from the Tools menu
- Specify global options
- Specify local options
- Examine output



520-TD-001-002
SSI&T 5-6

### Discussion Topics

**Step 2.** Specify any global options and/or a filename in which to save output.

To the prompt: global option(s) and list file?

- list out options (if any) that will be applied to all files being checked (see FORCHECK User Guide) and a list file to collect output.
- Here you may want to specify an output report file.
- If no global options and/or list file is desired, just press return.

**NOTE:** Information on global options may be found in the FORCHECK documentation.

The list file contains the source code as well as various summary statistics. See FORCHECK for further information.

**Step 3.** Specify local options and source filenames

To each prompt: local option(s) and file(s)?

- list out options (if any) and filenames. Any options listed will be applied only to the files immediately following.
  - Separate items in the list with spaces.

**NOTE:** Global options specified in step 2 will be applied to all files in addition to any local options listed in this step.


**Step 4.** Specify additional local options and source filenames.

- List options and files as in step 3.
- To end input, just hit return.

**Step 5.** Examine output or output file using desired text editor or display.

**NOTE:** Any action(s) following is determined by DAAC policy.

## Checking ANSI Compliance FORTRAN 90



**Purpose:**

To check that FORTRAN 90 source files are compliant with the ANSI standards

**Tool:**

- NAGWare FORTRAN 90 compiler

**Assumptions:**

- FORTRAN 90 source files are available, accessible, and have read permissions. Include files are available in the same directory as the Fortran 90 source files & have read permissions. The current shell being used is the C shell (csh). The NAGWare Fortran 90 compiler is accessible on the Science Processor (SGI Power Challenge).

520-TD-001-002

SSI&T 5-7

### Discussion Topics

**Purpose:** To check that Fortran 90 language source files are compliant with the ANSI standards.

**Assumptions:** The Fortran 90 source file(s) are available, accessible, and have read permissions. Any include files are available in the same directory as the Fortran 90 source files and have read permissions. The current shell being used is the C shell (csh). The NAGWare Fortran 90 compiler is accessible on the Science Processor (SGI Power Challenge).

Compliance checking for ANSI is done automatically via a NAGWare Fortran 90 compiler flag (unless it is explicitly turned off). Checking may be done as part of the science software build process or as a separate, preliminary step.

#### References:

SDP Toolkit User's Guide for the ECS Project, 333-CD-003-002


The Toolkit Primer, available through the World Wide Web:

URL: <http://newsroom.hitc.com/sdptoolkit/primer/tkprimer.html>

Data Production Software and Science Computing Facility (SCF) Standards and Guidelines, January 14, 1995 (CH-01 2/15/95), also on World Wide Web:

URL: <http://spsosun.gsfc.nasa.gov/SCFStand.html>

Checking ANSI Compliance  
FORTRAN 90 (cont'd)



- Select Xterm from SSIT Manager
- Telnet into the SGI
- Run the NAGWare FORTRAN 90 Compiler
- Review report file for non-compliance code

520-TD-001-002

SSI&T 5-8

### Discussion Topics

**Step 1.** Start a new shell on the SGI science processor within which to do the checking. Either: From the SSIT Manager GUI,

- choose the Tools menu. Then choose Xterm. Then telnet into the SGI.

or

In any currently available Xterm window,

- spawn a new Xterm session with: `xterm &`
- Then telnet into the SGI.

**NOTE:** Science software will be run on the SGI. Therefore, checking for ANSI via the compiler should be done on the same machine.

**Step 2.** Run the NAGWare Fortran 90.

ANSI checking is automatic unless it is explicitly turned off. Save the results to an output report file.

- Within the Xterm window, enter: `f90 -c *.f90 >& reportfile`

where `reportfile` is the filename under which to save the results.

Be certain that it is the NAGWare version of the Fortran 90 compiler being run. SGI's version should not be used.

- Type `f90 -v` (to verify this)

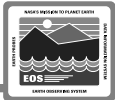
The `-c` flag causes only compilation and no linking.

**Step 3.** Review the report file for any non-compliant code.

- Visually scan the file using “more” or “cat”, or view the file using a preferred text editor (e.g. vi, emacs, xedit).
- Resolve any issues of non-compliance according to prescribed procedures.



## Checking ANSI Compliance C



**Purpose:**

- To check that C language source files are compliant with the ANSI standard

**Tool:**

- C compiler on the SGI Power Challenger

**Assumptions:**

- C source files are available, accessible, and have read permissions. Include files are available in the same directory as the C source files & have read permissions. The C compiler is accessible on the Science Processor (SGI Power Challenge).

520-TD-001-002SSI&T 5-9

### Discussion Topics

**Purpose:** To check that C language source files are compliant with the ANSI standard.

**Assumptions:** The C source file(s) are available, accessible, and have read permissions. Any include files are available in the same directory as the C source files and have read permissions. The C compiler is accessible on the Science Processor (SGI Power Challenge).

Compliance checking for ANSI is done via a C compiler flag.

Checking may be done as part of the science software build process or as a separate, preliminary step.

**References:**

cc man pages on the SGI

SDP Toolkit User's Guide for the ECS Project, 333-CD-003-002

The Toolkit Primer, available through the World Wide Web:

URL: <http://newsroom.hitc.com/sdptoolkit/primer/tkprimer.html>


ANSI C Programming Language Standard, X3.159-1989

IEEE Standard 1003.1, POSIX-Part 1: System Application Program Interface (API) [C Language]

Data Production Software and Science Computing Facility (SCF) Standards and Guidelines, January 14, 1995 (CH-01 2/15/95), also on World Wide Web:

URL: <http://spsosun.gsfc.nasa.gov/SCFStand.html>

### Checking ANSI Compliance C (cont'd)



- Select Xterm from SSIT Manager
- Telnet into the SGI
- Run the C Compiler with ANSI Checking Enabled
- Review report file for non-compliant code

520-TD-001-002

SSI&T 5-10

## Discussion Topics

**Step 1.** Start a new shell on the SGI science processor within which to do the checking. Either: From the SSIT Manager GUI,

- choose the Tools menu. Then choose Xterm.
- Then telnet into the SGI.

or

In any currently available Xterm window,

- spawn a new Xterm session with: `xterm &`
- Then telnet into the SGI.

NOTE: Science software will be run on the SGI. Therefore, checking for ANSI via the compiler should be done on the same machine.

**Step 2.** Run the C compiler with ANSI checking enabled and save the results to an output report file.

- Within the Xterm window, enter: `cc -c -32 -ansiposix *.c >reportfile`

where reportfile is the filename under which to save the results.

The `-c` flag causes only compilation and no linking.


The `-ansiposix` is the SGI compiler flag to enable ANSI checking.

The `-32` flag causes compilation in old 32-bit mode.

**Step 3.** Review the report file for any non-compliance code.

- Visually scan the file using `more` or `cat`, or view the file using a preferred text editor (e.g. `vi`, `emacs`, `xedit`).
- Resolve any issues of non-compliance according to prescribed DAAC procedures.

## Checking ANSI Compliance Ada



**Purpose:**

- To check that Ada language source files are compliant with the ANSI standards

**Tool:**

- Ada compiler on the SGI Power Challenger

**Assumptions:**

- Ada source files are available, accessible, and have read permissions. Ada software builds will be done using the (VADS)

520-TD-001-002

SSI&T 5-11

### Discussion Topics

**Purpose:** To check that Ada language source files are compliant with the ANSI standards.

**Assumptions:** The Ada source file(s) are available, accessible, and have read permissions. Ada software builds will be done using the Verdex Ada Development System (VADS) which provides a complete environment for building (and developing) Ada software.

**NOTE:** Ada compilers are subjected to a validation process by the DoD Ada Committee. Thus, any code successfully compiled by a validated compiler is, by definition, fully compliant.

The location of the VADS tools is accessible via the path environment variable on the SGI.

**References:**

U.S. Department of Defense, Ada Language Reference Manual, MIL-STD-1815-A

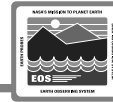
Data Production Software and Science Computing Facility (SCF) Standards and Guidelines, January 14, 1995 (CH-01 2/15/95), also on World Wide Web:

URL: <http://spsosun.gsfc.nasa.gov/SCFStand.html>

VADSelf, Silicon Graphics Computer Systems IRIX, Version 6.2.1, March 7, 1994, User's Guide, Rational Software Corporation

The VADS compiler complies fully with ANSI/MIL-STD-1815A-1983 (Ada RM).

### Checking ANSI Compliance Ada (cont'd)



- Select Xterm from SSIT Manager
- Telnet into the SGI
- Go into the directory where the source code resides
- Create a VADS library
- Run the Ada compiler

520-TD-001-002

SSI&amp;T 5-12

### Discussion Topics

**Step 1.** Start a new shell on the SGI science processor within which to do the checking.  
Either: From the SSIT Manager GUI,

- choose the Tools menu. Then choose Xterm.
- Then telnet into the SGI.

or

In any currently available Xterm window,

- spawn a new Xterm session with: `xterm &`
- Then telnet into the SGI.

**NOTE:** Science software will be run on the SGI. Therefore, checking for ANSI via the compiler should be done on the same machine.

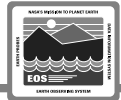
**Step 2.** `cd` to the directory where the source code resides.

**Step 3.** Create a VADS library directory.

All Ada compilation must occur in a VADS Ada library

- Enter `a.mklib`

## Checking Process Control File



**Purpose:**

- To check that Process Control Files (PCFs) are syntactically correct and contain all necessary info. for PGEs to run

**Tool:**

- SSI&T manager

**Assumptions:**

- SSI&T manager is running. PCFs are available, accessible, and have read permission

520-TD-001-002

SSI&T 5-13

### Discussion Topics

**Purpose:** To check that Process Control Files (PCFs) are syntactically correct and contain all necessary information for PGEs to run within the production environment.

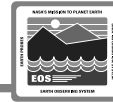
**Assumptions:** The SSIT Manager is running. The Process Control File(s) are available, accessible, and have read permissions.

#### References:

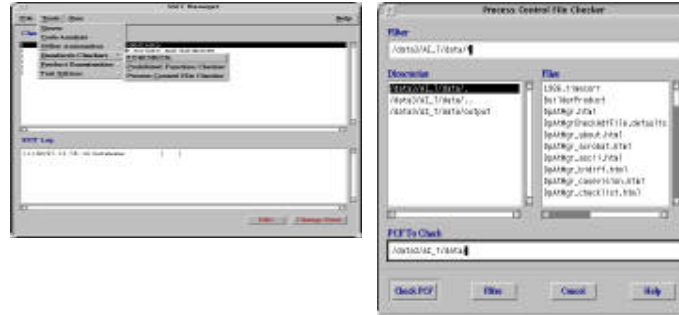
Interim Release One (Ir1) Maintenance and Operations Procedures, 609-CD-001-001

Help screen available through SSIT Manager (from SSIT Manager, choose Help)

## Checking Process Control Files (cont'd)



- **Invoke Process Control File (PCF) Checker**



520-TD-001-002

SSI&amp;T 5-14

## Discussion Topics

### Step 1. Invoke Process Control File Checker

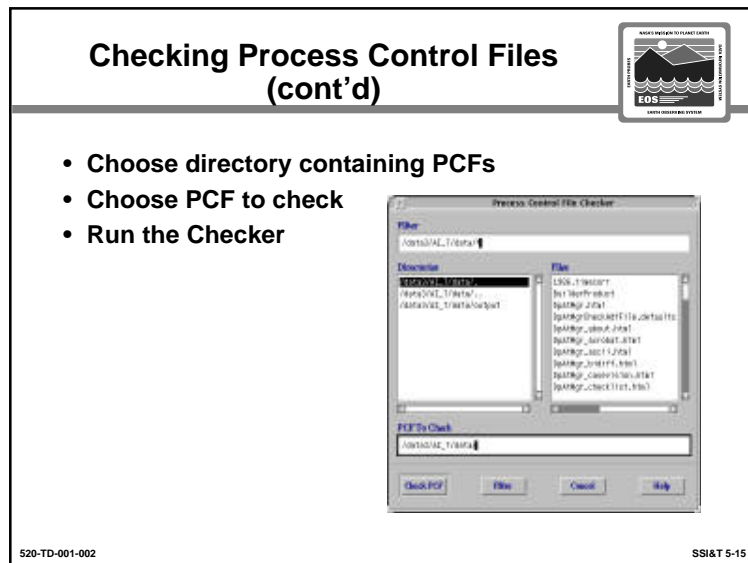
- From Tools menu, choose Standards Checkers.
- Then choose Process Control File Checker.

Process Control File Checker GUI will be displayed.

or

- Run the SDP Toolkit PCF checker from the command line
  - Type: `pccheck.sh`

**NOTE:** for more information consult the SDP Toolkit User's Guide



### Discussion Topics

#### Step 2 . Choose directory containing the Process Control File(s)

- Within the Directories subwindow, double-click on the desired directory.
- Repeat until the proper directory is displayed.

The files contained within the directory will be displayed in the Files subwindow according to the Filter defined in the Filter subwindow.

#### Step 3. Choose the Process Control File to check.

- Within the Files subwindow, highlight the Process Control File to check.

**NOTE:** Only one Process Control File can be checked at a time.

#### Step 4. Run the checker.

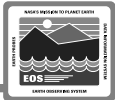
- Click on the button marked Check PCF.

PCF Checker Results GUI will be displayed. Errors and warnings found will be listed in the main subwindow.

#### Step 5. Optionally, save the results to a file or send to printer.

- To save the report to a file, click on the button labeled Save.
- To print the report, click on the button labeled Print.
- Choosing Save To File will bring up a Save GUI.
- Use this GUI to specify directory and filename under which to save the results.
- Choosing Print will bring up a Working GUI. Choosing the OK button will send the results to the default printer.

### Checking Metadata Control Files



**Purpose:**

- To check entries in the Metadata Control File (MCF) for each PGE generating a data product
- To check the MCF data for valid range values with the data dictionary

**Tool:**

- Text Editor

**Assumptions:**

- Each PGE has been checked for mandatory tool compliance by the SSI&T team

520-TD-001-002 SSI&T 5-16

## Discussion Topics

**Purpose:** To check entries in the Metadata Control File (MCF) for each PGE generating a data product, also checking the MCF data for valid range values with the data dictionary

**NOTE:** A system data dictionary was supplied with the Toolkit 5 delivery. Each instrument team had the option of adding to instrument specific information to the data dictionary if they desired.


**Assumptions:** The science software has been ingested into the I&T environment at the DAAC. Each PGE has been checked for mandatory tool compliance by the SSI&T team. A text editor is available to the SSI&T team.

### References:

1. SDP Toolkit User's Guide for the ECS Project, 333-CD-003-002
- 2 .The Toolkit Primer, available through the World Wide Web:  
URL: <http://newsroom.hitc.com/sdptoolkit/primer/tkprimer.html>
3. Science Data Processing Segment (SDPS) Data Design and Data Production Schema Specification for the ECS Project. Final Version for Release A. December, 1995. Document # 311-CD-002-004
4. Object Description Language Processing Software Library Manual - available through the following: "<http://stardust.jp1.nasa.gov/stdref.chap12.htm1>



### Checking Metadata Control Files (cont'd)



- Check MCF and Data Dictionary (DD) file names in the PCF
- Check contents of the MCF
- Correct MCF or DD errors

520-TD-001-002

SSI&T 5-17

## Discussion Topics

**Step 1.** Check for the filenames of the Metadata Control File (MCF) and (if included) Data Dictionary in the Process Control File submitted for each PGE.

- Use steps listed for “Checking Process Control Files”

**Step 2.** Check the contents of the MCF and additions to the DD if the instrument team added information to the system DD for Ir1 delivery.

- from SSIT manager select Tools menu
- From Tools select Xterm
- cd to the appropriate directory containing various source code for the PGEs
- start a text editor of your choice
- Load the desired file into the text editor

**NOTE:** You will be checking the contents of the MCF for such things as Mandatory Metadata using Appendix J of the SDP Toolkit Users Guide or SDPS database design and database schema specifications


**Step 3.** Correct the MCF or DD if errors such as Object Description Language (ODL) syntax or range values listed in the MCF or DD are invalid (e.g., typing mistakes are made).

- Contact the instrument team on how to correct errors found
- Correct errors if permission is granted by the instrument team

or

- have them correct the errors and send you a new package

## Checking for Prohibited Functions



**Purpose:**

- Check that science software contains no “Prohibited” functions, which might interfere with the production environment

**Tool:**

- Prohibited Function Checker

**Assumptions:**

- Science software code is available and has read permission

520-TD-001-002

SSI&T 5-18

### Discussion Topics

#### **Purpose:**

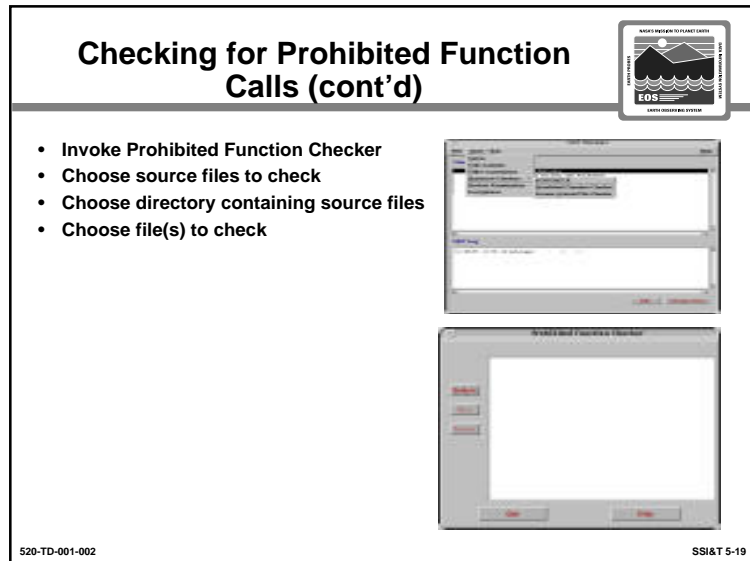
- Check that science software contains no “Prohibited” functions, which might interfere with the production environment

#### **Tool:**

- Prohibited Function Checker

#### **Assumptions:**

- Science software code is available and has read permission



### Discussion Topics

#### Step 1. Invoke Prohibited Function Checker.

- From Tools menu of the SSIT Manager choose Standards Checkers.
- Then choose Prohibited Function Checker.

The Prohibited Function Checker GUI will be displayed.

or

- type: "badfunc" or "xbadfunc" from the command line

#### Step 2. Choose the source files to check.

- Click on the button labeled Analyze.

The File Selector GUI will be displayed.

#### Step 3. Choose directory containing the source files.

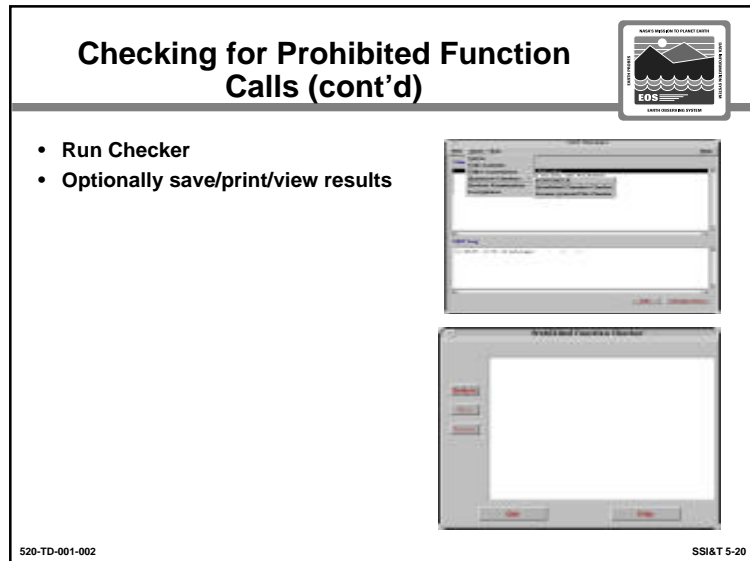
- Within the Directories subwindow, double-click on the desired directory.
- Repeat until the proper directory is displayed.
  - The files contained within the directory will be displayed in the Files subwindow.

#### Step 4. Choose the file(s) to check.

- Within the Files subwindow highlight the files to check.
- Choose groups of contiguous files by holding down and dragging the left mouse button.
- Choose non-contiguous files by holding down the Control key while clicking on the filenames.

The selected files will be displayed in the Selections subwindow.

**NOTE:** files from only one directory at a time can be done. Choosing another directory causes any filenames listed in this subwindow to disappear.



## Discussion Topics

### Step 5. Run the checker.

- Click on the button labeled Ok.

The File Selector GUI will disappear. The files chosen will be displayed in the Prohibited Function Checker window as they are checked.

### Step 6. Check the results.

- Click on the button labeled Report.

The Report GUI will be displayed. For each file, a list of prohibited functions found will be displayed.

### Step 7. Optionally, save the results to a file or send to printer.

- To save the report to a file, click on the button labeled Save.
- To print the report, click on the button labeled Print.

Choosing Save To File will bring up a Save GUI. Use this GUI to specify directory and filename under which to save the results. Choosing Print will send the results to the default printer.

### Step 8. Optionally, view the occurrences of prohibited functions in the source files.

- In the Prohibited Function Checker GUI, highlight one of the source files listed.
- Then click on the button labeled "View".

The Source Code GUI will be displayed. Occurrences of prohibited functions found will be highlighted. The button labeled "Next" will show successive occurrences of highlighted prohibited. (The "Next" button does not bring in the next file.) The "Done" button removes the Source Code GUI.

### Step 9. Quit the Prohibited Function Checker.

- In the Prohibited Function Checker GUI, click on the button labeled Quit. The Prohibited Function Checker GUI will disappear.